UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

#15

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/637,015	08/14/2000	Gopal S. Krishna	95-319	8307

20736 7590 02/23/2004

MANELLI DENISON & SELTER
2000 M STREET NW SUITE 700
WASHINGTON, DC 20036-3307

EXAMINER	
MOORE, IAN N	
ART UNIT	**PAPER NUMBER**
2661	8

DATE MAILED: 02/23/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No. 09/637,015	Applicant(s) KRISHNA ET AL.
	Examiner Ian N Moore	Art Unit 2661

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-16 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-16 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed February 4, 2004 have been fully considered but they are not persuasive.

Regarding claims 1-16, the applicant alleged, "Deb'990 neither discloses or suggests storing a plurality of templates configured for identifying respective data formats, each template having at least one min term configured for comparing a corresponding prescribed value to a corresponding selected byte of the incoming data packet".

In response to the applicant's argument, Deb'990 discloses storing a plurality of templates configured for identifying respective data formats (see Fig. 3A Programming instruction set; and col. 11, line 13-14 and 43-48; noted that the user may configure a software instruction set by designating a specific instruction to be performed on in-coming packet. These instructions are set to identify for various type of data format (i.e. IP, TCP, and SNMP type). Therefore, it is clear that instruction identifies the type of data format of the incoming packet.), each template having at least one min term (see Fig. 3A, the user defined data structure type and contents; see col. 12, line 46-56; noted that per specification (page 6, line 17 to page 7, line 4), "min term" has a function of defining a specific data format. The "data structure type and contents" also has a function of defining a specific data format. Therefore, it is clear that data structure type and content is the "min term".) configured for comparing a corresponding prescribed value to a corresponding selected byte of the incoming data packet (see Fig. 3B, Analyzing Computer 337; and col. 13, line 4-35, noted that the user defined contents are loaded/stored into the CAM 334, RAM 302, Comparators 336. A

selected word/field (i.e. a word/field = a byte) from the incoming packet stream is being compared to the user-defined instruction set. Moreover, since the user instruction set/content includes a "field" for each data type (i.e. TCP header field, IP header field, Application header field, IP destination address field, and etc.) of the incoming packet, it is clear that the comparing process must be done between the field/word of the incoming packet and respective field/word of the instruction set stored in the memory.); also see Fig. 4A method step 402; Fig. 4B; and col. 25, line 20-29. Also, it is noted that applicant's remarks detailed the specific functions of invention compared to Deb'990, it is clear that applicant's **broad** claim limitations still read on Deb'990.

The applicant alleged, " ...Deb'990's micros-RISC stream processor cannot simultaneously compare the selected byte to the min terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port..."

In response to the applicant's argument, Deb'990 discloses a micros-RISC stream processor simultaneously compare the selected byte to the min terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port (note that a byte/word of received packet from the physical layer/port is examined/compared against the pre-defined/programmed/stored instructions sets (i.e. min terms). The examining/comparing process is performed in-line while the process of streaming/transmitting packets to the upper layer. Thus, it is clear the examining/comparing process must be performed **at the same time** (i.e. simultaneously) as streaming/transmitting process. Moreover, the examining/comparing process begins at the initial position of the received packet. Thus, it is clear that the process must start as soon as (i.e. immediately) the

packet arrives at the physical layer/port. Also, note that each pipeline register stage segments/defines each received data word/byte to perform the examining/comparing process in **parallel manner** in order to reduce the processing time. For example, Stage 1 contains 57^{th} data word, stage 2 contains 56^{th} data word and stage 3 contains 55^{th} data word. The examining/comparing process is performed for these data words at <u>one time</u>. Thus, it is clear that the examining/comparing process is performed **"at the same time/simultaneously"** rather than "sequentially". See col. 4, lines 3-16, and col. 13, lines 35-50).

The applicant alleged, " ...Deb'990 does not teach generating a frame tag based on the comparison result as soon as a last bit of the data packet is received at the network switch port..."

In response to the applicant's argument, Deb'990 discloses generating a frame tag (see Fig. 8, Encapsulation header 804 is a frame tag); based on the comparison result as soon as a last bit of the data packet is received at the network switch port (see col. 29, line 10-29; and col. 21, line 30-46; noted that a new tag header is generated with an appended index (i.e. the user defined instructions) based upon compared/determined output data. The purpose of Deb'990 invention is to increase transmit and received packet processing rates while reducing a host CPU' s processing burden; see col. 3, lines 55-59. Deb'990 achieves the purpose by examining/comparing at the physical layer upon receiving the packet at the same time, as creating the examined/compared/outputted results that the identifies the type of packet and creating a header in order to reduce the upper layer processing burden. Also, Deb'990 discloses that the processing is performed **on-the-fly at line rate;** see col. 4, lines 49-63. Thus, it is clear that a header (i.e. a frame tag) is must be generated according to the

examined/compared results on-the-fly according to the line rate (i.e. as soon as a last bit of the data packet is received at the physical layer switch port). In order to achieve the line rate processing (i.e. no processing delay whatsoever), parallel and simultaneous examining/comparing must be performed, and a header must be produced based upon examined/compared result as soon the last bit of the packet is being received at the physical layer. Thus, when performing on-the-fly at line rate, it reduces the transmission delay, since the method and apparatus increases the speed of examining/comparing, increases the process of generating results, and increases the process of creating of a header.

In view of the above, the examiner believes that the reference, Deb'990, as set forth in the 102 rejections of **claims 1-16** is proper since the broadly claimed limitations still read on the prior art.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

1. Claim 1-6, 8-9, and 11-15 are rejected under 35 U.S.C. 102(e) as being anticipated by Deb (U.S. 6,172,990).

Regarding Claim 1, Deb '990 discloses a method of evaluating an incoming data packet at a network switch port, the method comprising:

storing a plurality of templates configured for identifying respective data formats (see Fig. 3A Programming instruction set; and col. 11, line 13-14 and 43-48; noted that the user may configure a software instruction set by designating a specific instruction to be performed on in-coming packet. These instructions are set to identify for various type of data format (i.e. IP, TCP, and SNMP type). Therefore, it is clear that instruction identifies the type of data format of the incoming packet.), each template having at least one min term (see Fig. 3A, the user defined data structure type and contents; see col. 12, line 46-56; noted that per specification (page 6, line 17 to page 7, line 4), "min term" has a function of defining a specific data format. The "data structure type and contents" also has a function of defining a specific data format. Therefore, it is clear that data structure type and content is the "min term".) configured for comparing a corresponding prescribed value to a corresponding selected byte of the incoming data packet (see Fig. 3B, Analyzing Computer 337; and col. 13, line 4-35, noted that the user defined contents are loaded/stored into the CAM 334, RAM 302, Comparators 336. A selected word/field (i.e. a word/field = a byte) from the incoming packet stream is being compared to the user defined instruction set. Moreover, since the user instruction set/content includes a "field" for each data type (i.e. TCP header field, IP header field, Application header field, IP destination address field, and etc.) of the incoming packet, it is clear that the comparing process must be done between the field/word of the incoming packet and respective field/word of the instruction set stored in the memory.); **also see Fig. 4A method step 402; Fig. 4B; and col. 25, line 20-29**;

simultaneously comparing to the selected byte (see Fig. 3B, Analyzing Computer 337; col. 13, line 36-50; col. 4, lines 3-16, col. 13, lines 35-50, note that a byte/word of

received packet from the physical layer/port is examined/compared against the pre-defined/programmed/stored instructions sets (i.e. min terms). The examining/comparing process is performed in-line while the process of streaming/transmitting packets to the upper layer. Thus, it is clear the examining/comparing process must be performed **at the same time** (i.e. simultaneously) as streaming/transmitting process. Each word/field is being processed in parallel format utilizing pipeline register stages. Also, note that each pipeline register stage segments/defines each received data word/byte to perform the examining/comparing process in **parallel manner** in order to reduce the processing time. For example, Stage 1 contains 57^{th} data word, stage 2 contains 56^{th} data word and stage 3 contains 55^{th} data word. The examining/comparing process is performed for these data words at one time. Thus, it is clear that the examining/comparing process is performed "**at the same time/simultaneously**" rather than "sequentially".), the min terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port (see col. 13, line 51 to col. 14, line 6; noted that a selected word from an in-coming packet is being compared/examined according to the user defined data structure type and contents at the analyzing computer. The examining/comparing process begins at the initial position of the received packet. Thus, it is clear that the process must start as soon as (i.e. immediately) the packet arrives at the physical layer/port.); **also see Fig. 4A method step 404 and 406; Fig. 4C; and col. 25, line 31-32**;

generating a comparison result that identifies the incoming data packet, based on the comparisons of the min terms to the data bytes of the entire packet received by the network switch port (see col. 14, line 13-61; noted that compared/computed output (i.e. in the form of

the data structure) between selected word and the user defined data instructions are produced at the analyzing computer. The data structure contains the information regarding computed results such as TCP header field, IP header field, the source and destination addresses, and etc. Therefore, it is clear the "comparison result" is the "computed output".); **Also see Fig. 4A method step 408; and col. 25, line 33-35**; and

generating a frame tag (see Fig. 8, Encapsulation header 804 is a frame tag); based on the comparison result as soon as a last bit of the data packet is received at the network switch port (see col. 29, line 10-29; and col. 21, line 30-46; noted that a new tag header is generated with an appended index (i.e. the user defined instructions) based upon compared/determined output data; Also, Deb'990 discloses that the processing is performed **on-the-fly at line rate;** see col. 4, lines 49-63. Thus, it is clear that a header (i.e. a frame tag) is must be generated according to the examined/compared results on-the-fly according to the line rate (i.e. as soon as a last bit of the data packet is received at the physical layer switch port). **see Fig. 4A method step 408; and see col. 26, line 41-44**.

Regarding Claim 2, Deb '990 discloses the simultaneously comparing step includes:

loading the min terms (see Fig. 4A, step 402; and col. 16, line 15-19) corresponding to a first of the data bytes into a min term generator (see Fig. 4A, step 404; and see col. 16, line 24-39; noted that the user defined instruction/context are loaded/stored in the RAM, CAM, and Comparators associating to the initial byte of an incoming packet. Also, Analyzing Computer 337 in Fig. 3B has a functionality of a min term generator);

comparing in parallel the min terms loaded in the min term generator with the first of the data bytes (see Fig. 4A, step 406; col. 16, line 47-52; since the incoming packet are stored in the pipe line register stages in order, the first word (i.e. word count "0") is used when examining/comparing the received packet); and

outputting comparison results for the min terms loaded in the min term generator to an evaluation core (see Fig. 4A, step 408; col. 16, line 55-61; noted that after comparing/determining, the output from compared/determined data based upon comparisons are transferred to the Mux 318 and data transfers register 316. Thus, an evaluation core is a combined system of Analyzing Computer 337, Mux 318, and Data Structure Register 316 of Fig. 3B.).

Regarding Claim 3, Deb '990 discloses the simultaneously comparing step further includes loading the min terms corresponding to a second of the data bytes, contiguously following the first of the data bytes, into the min term generator (see Fig. 4C; col. 25, line 39-58; and col. 13, line 34-50; noted that a second of the data byte is stored contiguously according to the method defined in Fig. 4C).

Regarding Claim 4, Deb '990 discloses outputting the frame tag to a switch fabric (see Fig. 2A, a combined system of Tx micro-RISC Stream Processor 114a and Switch Table Lookup 806) configured for selectively switching the incoming data packet based on the corresponding frame tag (see col. 21, line 14-61; and col. 22, line 46-57; noted that the user defined appendix is append to an incoming packet, transferred to a lookup switch table,

outputted to a transmit processor to encapsulate the tag, and then routed/switched accordingly.)

Regarding Claim 5, Deb '990 discloses the storing step includes storing each min term in a memory as a table entry (see Fig. 3A, RAM 302, CAM 334, and Comparators 336),

each table entry having a location in the memory based on a location of the corresponding selected byte in the incoming data packet (see col. 13, line 24-35; noted that the user instructions are resident in word count 308 is configured to identify a desired word count in an in-coming packet. Therefore, it is clear that there is a location in the memory regarding a selected word so that the system can correctly determine/compare it according to the user defined instruction stored in the memory),

the table entry including a min term expression portion specifying the corresponding prescribed value and a comparison operand (see Fig. 3A, Data structure content field; and col. 11, line 43-56; noted that a data structure content field (i.e. pointer, data, and/or other) is used to identify what determination/operation will be performed), and

a template identifier field that specifies the templates that use the corresponding min term (see Fig. 3A, Data structure type field and col. 11, line 43-56; noted that a data structure type field (i.e. a Standard data, flag, or other fields) is used to identify what type of the user defined instruction will be used for comparison operation.)

Regarding Claim 6, Deb '990 discloses the generating step includes:

temporarily storing results of the comparisons of the min terms to the selected bytes of the incoming data packet (see col.14, line 28-30; noted that the compared/computed output data are stored in a data structure having a pointer to the currently selected word.)

detecting at least one matched template from the plurality of templates based on the results of the comparisons of the min terms and generating the comparison result based on the detected at least one matched template (see col. 13, line 17-23; a match signal is produced (i.e. match found) after matching according to the CAM 334 look up table which stored plurality of user defined instructions and the corresponding entry (i.e. compared/determined output data) are outputted.)

Regarding Claim 8, Deb '990 discloses the first of the data bytes corresponds to a first of the data bytes of a packet having a prescribed format, the simultaneously comparing step including evaluating the selected data byte relative to a beginning of the packet having the prescribed format (see col. 13, line 51 to col. 14, line 6; noted that the determination is performed by the analyzing computer from the initial word (i.e. beginning of the packet) and continues consecutively).

Regarding Claim 9, Deb '990 discloses the prescribed format is Internet protocol (IP) format (see Fig. 9, IP switching 3 – IP header 910 and data 906; and col. 20, line 54-60).

Regarding Amended Claim 11, Deb '990 discloses a network switch port filter configured for evaluating an incoming data packet, comprising:

a min term memory (see Fig. 3; RAM 302, CAM 334 and Comparators 336 which stored Programming instruction set) configured for storing min term values (see col. 11, line 13-14 and 43-49; noted that the user may configure the software instruction set by designating a specific instruction to be performed on in-coming packet. These instructions are set to identify the various type of data format (i.e. IP, TCP, and SNMP type). Therefore, it is clear that the instruction set identifies the type of data format of the incoming packet), each min term value (see Fig. 3A, data structure type and contents; and col. 11, line 18-20; noted that per specification (page 6, line 17 to page 7, line 4), "min term" has a function of defining a specific data format. The "data structure type and contents" also has a function of defining a specific data format. Therefore, it is clear that data structure type and contents is the "min term".) stored based on a location of a corresponding selected byte of the incoming data packet for comparison (see Fig. 3B, Analyzing Computer 337; and col. 13, line 4-35, noted that the user defined contents are loaded/stored into the CAM 334, RAM 302, and Comparators 336. A selected word/field (i.e. a word/field = a byte) from the incoming packet stream is being compared to the user defined instruction set/contents. Moreover, since the user instruction set/content includes a "field" or "word" for each data type (i.e. TCP header filed, IP header filed, Application layer header field, IP source/destination address field, and etc.) of the incoming packet, it is clear that the comparing process must be done between a word/field of the incoming packet and respective field/word of the instruction field stored in the memory.), a min term portion specifying a corresponding comparison operation (see Fig. 3A, Data structure content field and col. 11, line 43-56; noted that a data structure content field (i.e. pointer, data, and/or other) is used to identify what determination/operation will be

performed), and an equation identifier field that specifies templates that use the corresponding min term (see Fig. 3A, Data structure type field and col. 11, line 43-56; noted that a data structure type field (i.e. a Standard data, flag, or other fields) is used to identify what type of the user defined instruction will be used for comparison operation);

a min term generator (see Fig. 3B, Analyzing Computer 337; noted that analyzing computer performs both equation core and min term generating function.) configured for simultaneously comparing a byte of the incoming data packet (col. 13, line 36-50; col. 4, lines 3-16, col. 13, lines 35-50, note that a byte/word of received packet from the physical layer/port is examined/compared against the pre-defined/programmed/stored instructions sets (i.e. min terms). The examining/comparing process is performed in-line while the process of streaming/transmitting packets to the upper layer. Thus, it is clear the examining/comparing process must be performed **at the same time** (i.e. simultaneously) as streaming/transmitting process. Each word/field is being processed in parallel format utilizing pipeline register stages. Also, note that each pipeline register stage segments/defines each received data word/byte to perform the examining/comparing process in **parallel manner** in order to reduce the processing time. For example, Stage 1 contains 57th data word, stage 2 contains 56th data word and stage 3 contains 55th data word. The examining/comparing process is performed for these data words at one time. Thus, it is clear that the examining/comparing process is performed "**at the same time/simultaneously**" rather than "sequentially".) immediately upon receipt of the incoming data byte (see col. 13, line 51 to col. 14, line 6; noted that a selected word from an in-coming packet is being compared/examined according to the user defined data structure type and contents at the analyzing computer. The

examining/comparing process begins at the initial position of the received packet. Thus, it is clear that the process must start as soon as (i.e. immediately) the packet arrives at the physical layer/port.) with the min terms that correspond to the received byte and generating respective min term comparison results (see col. 14, line 34-61; noted that the compared/computed output (i.e. in the form of the data structure) between selected word and the user defined data instructions are produced at the analyzing computer. The data instruction contains the information regarding the computed output such as TCP header field, IP header field, the source and destination addresses, and etc. Therefore, it is clear that the "comparison results" is the "computed/determined output".); And

an equation core (see Fig. 3B, Analyzing Computer 337; noted that analyzing computer performs both equation core and min term generating function.) configured for generating a frame tag identifying the incoming data packet based on the min term comparison results relative to the templates (see col. 29, line 10-29; and col. 21, line 30-46; noted that a new tag header is generated with an appendix index (i.e. the user defined contents) based upon compared/computed output data; Also, Deb'990 discloses that the processing is performed **on-the-fly at line rate;** see col. 4, lines 49-63. Thus, it is clear that a header (i.e. a frame tag) is must be generated according to the examined/compared results on-the-fly according to the line rate (i.e. as soon as a last bit of the data packet is received at the physical layer switch port).

Regarding Claim 12, Deb '990 discloses a frame identifier (see Fig. 3B, CAM 334) configured for identifying a type of layer 2 packet (see col. 9, line 18-23; and col. 13, line 16-

23; noted that MAC layer is a Layer 2, and the CAM identifies by utilizing a look up table to process for each word type before passing over to the higher layers), the selected byte of the incoming data packet determined based on the identified type of layer 2 packet (see col. 13, line 59-63; col. 19, line 59 to col. 20, line 3; and Fig. 5A and 5B; noted that selected word of the incoming packet is determined according to the user defined instructions for different type of layer 2.)

Regarding Claim 13, Deb '990 discloses the location of each stored min term value is relative to a beginning of an IP frame (see Fig.9, IP header 910; and col. 20, line 54-60; noted that received packet header is an IP header, and therefore, it is an IP frame.) within the layer 2 packet (see col. 13, line 33-50; the user defined instructions are resident in word count 308, and it is configured to identify a desired word count in an in-coming packet. Each time a new packet is received by micro-RISC stream processor 114a, a word counter 307 will reset to "0", and then word counter 307 begins sequentially counting each word that is received into pipeline register stages 323 from data path 115a. Therefore, each stored the user defined instruction is relative to a beginning of an IP frame).

Regarding Claim 14, Deb '990 discloses a min term controller (see Fig. 3B, Execution Logic 312) configured for fetching the min terms from the min term memory corresponding to a selected byte of the IP frame within the incoming data packet (see col. 15, line 45-49 and col.14, line 12-19; noted that the execution logic unit is preferably designed to control the examination of the received packet by the analyzing computer.)

Regarding Claim 15, Deb '990 discloses the equation core generates the frame tag at a wire rate of the incoming data packet and prior to an end of the incoming data packet (see col. 11, line 21-42; col. 21, line 30-46; and col. 29, line 10-27; Noted that both Tx and Rx micro-RISC stream processors are considered as one system. Therefore, the processor has a capability of generation a frame tag, which encapsulates to the data frame.)

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claim 7 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Deb '990 in view of Connery (U.S. Patent 6,570,884).

Regarding claim 7, Deb '990 discloses generating one final frame tag when one template matches the incoming data packet as described above in Claim 1, 5, 6.

Deb '990 does not explicitly disclose including resolving a priority of templates (see Connery '884 Fig. 3, Pattern Match units 1-4) to one final template when more than one template matches the incoming data packet (see Connery '884 col. 7, line 52-62; noted that when there are multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format).

However, this limitation is taught by Connery '884. Noted that Deb '990 teaches generating a final tag value after matching the words to the user-defined instructions. Deb '990 further teaches prioritizing the incoming data traffic at the receiver utilizing various buffers (see Deb '990 col. 10, line 59-65) and transmitting different type of traffic according to the priority (see Deb '990 col. 7, line 33-38). Deb '990 also teaches the user-defined instruction, and the user is able to define/prioritize the traffic type (i.e. voice vs. data or IP vs. SNMP) for by utilizing instruction. Connery '884 teaches prioritizing/finalizing/defining the final match if there is more than one match. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Deb '990 as taught by Connery '884 for the purpose of the hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, see Connery '884 col. 3, line 46-53. The motivation being that by determining/finalizing a match pattern, it can minimize the probability of faulty matches.

Regarding claim 16, Deb '990 discloses a tag generator device generating a final frame tag value after a given template matches the incoming data packet as described above in claim 11.

Deb '990 does not explicitly disclose a priority device (see Connery '884 Fig. 3, Processor 220) configured for resolving a priority of templates (see Connery '884 Fig. 3, Pattern Match units 1-4) to one final frame template when more than one template matches the incoming data packet (see Connery '884 col. 7, line 52-62; noted that when there are

multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format).

However, this limitation is taught by Connery '884. Noted that Deb '990 teaches a tag generator device, which generates a final tag value after matching words to the user defined instruction. Deb '990 further teaches prioritizing the incoming data traffic at the receiver utilizing various buffers (see Deb '990 col. 10, line 59-65) and transmitting different type of traffic according to the priority (see Deb '990 col. 7, line 33-38). Deb '990 also teaches the user-defined instruction (i.e. templates), and the user is able to define/prioritize the traffic type (i.e. voice vs. data or IP vs. SNMP) for by utilizing the instruction/templates. Connery '884 teaches a processor device, which prioritizes/finalizes the final match. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Deb '990 as taught by Connery '884 for the purpose of the hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, see Connery '884 col. 3, line 46-53. The motivation being that by determining/finalizing a match pattern, it can minimize the probability of faulty matches.

3. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Deb '990 in view of Bellenger (U.S. 5,802,054).

Regarding Claim 10, Deb '990 discloses the step of generating the comparison result based on the detected at least one matched template includes: identifying for each of the min

terms compared with the incoming data packet and specifying a unique result for a selected group of the templates; and generating the comparison result by having the detected at least one matched template as described above in Claim 1, 5, and 6.

Deb '990 does not explicitly disclose an equation (see Bellenger '054 col.13, line 61 to col.14, line 39; noted that a template register (i.e. an equation) where each bit specifics one byte of each header.), each equation specifying a unique result for a selected group of the templates; and generating the comparison result by the equation (see col. 14, line 38 to col. 15, line 10; see also Fig. 6; noted that there are plurality of registers (i.e. equations) loaded with templates, and each register has a specific filtering functions depending on many protocol possibilities (i.e. the second register detects a specific protocol type, and the third register detects a hierarchy number, and etc.) Then, the multiplexer multiplexes each resulted/selected-filtered data into one unique result and outputted.)

However, this limitation is taught by Bellenger '054. Noted that Deb '990 teaches generating comparison results per the user defined types and instructions. Bellenger '054 discloses plurality of template registers (i.e. equations) performing parallel processing in order to obtain unique result. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the system of Deb '990 as taught by Bellenger '054 for the purpose of utilizing a logic which identifies an incoming frame and generates a value very quickly; thus, by allowing for cut through of frames in a switch node, a transmission of a frame on an outgoing port can begin before the complete frame has been received at the incoming port. Therefore, a high bandwidth and very flexible network switch is achievable according to the present invention with a simple, scalable, low-cost

architecture, see Bellenger '054 col. 4, line 23-39 and col. 17, line 5-7. The motivation being that by performing simultaneous/cut through filtering function, it can increase flexibility of the switch and increase the bandwidth of the transmission.

Notes/Remarks

4. Claim rejections based on 35 USC § 112 second paragraph for Claims 5 and 11 are withdraw since the claims are amended accordingly.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ian N Moore whose telephone number is 703-605-1531. The examiner can normally be reached on M-F: 9-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ken Vanderpuye can be reached on 703-308-7828. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

INM
2/20/04

KENNETH VANDERPUYE
PRIMARY EXAMINER